# 1 Setting

## 1.1 vimrc

```
set ts=4 sts=4 sw=4
set ai si nu
```

# 2 Math

## 2.1 basic arithmetic

```
typedef long long ll;
typedef unsigned long long ull;

// calculate ceil(a/b)
// |a|, |b| <= (2^63)-1 (does not dover -2^63)
ll ceildiv(ll a, ll b) {
    if (b < 0) return ceildiv(-a, -b);
    if (a < 0) return (-a) / b;
    return ((ull)a + (ull)b - 1ull) / b;
}

// calculate floor(a/b)
// |a|, |b| <= (2^63)-1 (does not cover -2^63)
ll floordiv(ll a, ll b) {
    if (b < 0) return floordiv(-a, -b);
    if (a >= 0) return a / b;
    return -(ll)(((ull)(-a) + b - 1) / b);
}

// calculate n^k % m
ll modpow(ll n, ll k, ll m) {
    ll ret = 1;
    n %= m;
    while (k) {
        if (k & 1) ret = ret * n % m;
        n = n * n % m;
        k /= 2;
    }
    return ret;
}

// range modular inverse
int modinv[SIZE];
void calc_range_modinv(int n, int mod) {
    modinv[1] = 1;
    for (int i = 2; i <= n; ++i)
        modinv[i] = (ll)(mod - mod/i) * modinv[mod%i] % mod;
}
```

## 2.2 euler totient function

## 2.3 chinese remainder theorem

# 3 Data Structure

## 3.1 fenwick tree